ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRẦN THỊ THÚY HẰNG

PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG TƯƠNG TÁC GIAO DIỆN NGƯỜI DÙNG CHO CÁC ỨNG DỤNG WEB

Ngành: Công nghệ thông tin

Chuyên ngành: Kỹ thuật phần mềm

Mã Số: 62 48 01 03

TÓM TẮT LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2016

MỤC LỤC

Chương 1: Giới thiệu

Các ứng dụng Web được phát triển một cách mạnh mẽ nhưng hầu như không đáp ứng được các nhu cầu của khách hàng. Làm thế nào để đảm bảo được chất lượng của các ứng dụng Web là vấn đề quan trọng trong phát triển phần mềm. Kiểm thử gần như là phương pháp duy nhất để đảm bảo chất lượng cho các sản phẩm phần mềm. Kiểm thử viên phải thực hiện một cách lặp đi lặp lại một bộ các ca kiểm thử để nhằm xem xét phần mềm có xảy ra lỗi hay không mặc dù đội phát triển chỉ thực hiện sửa một phần nhỏ trong hệ thống. Các kỹ thuật hay phương pháp kiểm thử tự động được biết đến như là các giải pháp tiềm năng để giải quyết các vấn đề nêu trên. Những kỹ thuật này có thể phát hiện ra tất cả các lỗi có thể ứng dụng bằng cách tạo ra và thực hiện một cách tự động các ca kiểm thử. Điều này khiến chúng ta có thể cắt giảm đi rất nhiều thời gian và chi phí không cần thiết.

Luận văn này nghiên cứu phương pháp kiểm thử tự động tương tác người dùng của ứng dụng Web. Phương pháp này là một trong những phương pháp kiểm thử đang được áp dụng và sử dụng ngày càng rộng rãi trong việc kiểm thử các sản phẩm phần mềm nói chung cũng như các ứng dụng Web nói riêng. Trong trường hợp, chúng ta có một kịch bản của một ứng dụng Web, làm thế nào để kiểm tra tính đúng đắn của việc cài đặt có đúng theo như thiết kế ban đầu hay không? Việc kiểm tra này khó để thực hiện một cách tự động. Phương pháp kiểm thử tự động tương tác giao diện người dùng có ba cách chính: kiểm thử thủ công, kiểm thử bằng cách chụp và phát lại (capture and replay), kiểm thử dựa trên mô hình. Trong phạm vi nghiên cứu này, tôi tập trung nghiên cứu kiểm thử tương tác giao diện người dùng dựa trên mô hình nhằm đảm bảo hệ thống được cài đặt đúng theo thiết kế. Từ mô hình đặc tả hành vi của ứng dụng Web - mô hình ô-tô-mát hữu hạn trạng thái từ đó xây dựng một đồ thị có hướng để biểu diễn các trường hợp kiểm thử. Các ca kiểm thử chính là các nhánh trong đồ thị. Luận văn tập trung áp dụng và phát triển các công cụ hỗ trợ cho giai đoạn sinh bộ đầu vào cho công cụ sinh bộ kiểm thử. Từ bộ đầu vào, công cụ sinh các ca kiểm thử sẽ kiểm tra tất cả các nhánh của mô hình ô-tô-mát hữu hạn trạng thái và kiểm thử hệ thống một cách tự động.

Nội dung của luận văn được trình bày trong năm chương và phần kết luận. Chương 1 giới thiệu về đề tài, chương này trình bày các ngữ cảnh, những lý do chọn đề tài, mục tiêu của đề tài và cấu trúc của luận văn. Chương 2 trình bày về kiểm thử tự động, so sánh giữa kiểm thử tự động và kiểm thử thủ công, các kiểu kiểm thử tự động, nếu lý thuyết của kiểm thử tự động dựa trên mô hình. Chương 3 mô tả phương pháp đặc tả tương tác giao diện cho các ứng dụng web, trong chương này người viết có nếu cách đặc tả, xây dựng mô hình đặc tả, và cách biểu diễn mô hình đặc tả như thế nào. Chương 4 mô tả về việc sinh và thực thi các ca kiểm thử tự động và ví dụ áp dụng. Chương 5 giới thiệu công cụ và trình bày kết quả thực nghiệp vào một ứng dụng Web của FPT. Cuối cùng là phần kết luận, định hướng mở rộng và tài liệu tham khảo.

Chương 2: Kiểm thử phần mềm tự động

2.1 Kiểm thử phần mềm thủ công và Kiểm thử phần mềm tự động

Phân biệt Kiểm thử tự động và kiểm thử thủ công. Mỗi một loại kiểm thử đều có tính ưu và nhược điểm khác nhau. Dựa vào các yếu tố này để nếu bật lợi ích của kiểm thử tự động

2.2 Các kiểu kiểm thử tự động và kiểm thử giao diện người dùng

2.2.1 Các kiểu kiểm thử tự động

Theo tài liệu [4], dựa trên mô hình vừa nêu chúng ta có thể chia kiểm thử tự động thành ba kiểu:

- Kiểm thử mức đơn vị (Unit test)
- Kiểm thử tích hợp (có thể là tích hợp các mô đun hoặc tích hợp hệ thống)
- Kiểm thử giao diện người dùng

Có rất nhiều công cụ kiểm thử tự động có thể áp dụng cho từng mức kiểm thử. Trong nghiên cứu của luận văn này, người viết muốn đề cập đến kiểu kiểm thử giao diện người dùng. Đây là kiểu kiểm thử được áp dụng phổ biến nhất trong việc tương tác với hệ thống.

2.2.2 Kiểm thử tương tác giao diện người dùng

Một giao diện người dùng là một phần quan trọng của tương tác hệ thống. Người dùng có thể dựa trên các giao diện để có thể hiểu hệ thống như thế nào và quyết định sử dụng hay không sử dụng. Nói về tương tác giao diện người dùng chúng ta có ba cách chính: kiểm thử thủ công, kiểm thử bằng cách chụp và phát lại (capture and replay), kiểm thử dựa trên mô hình. Trong phạm vi nghiên cứu này, người viết tập trung nghiên cứu kiểm thử tương tác giao diện người dùng dựa trên mô hình - một trong những phương pháp đang phát triển hiện nay.

2.3 Kiểm thử tự động dựa trên mô hình

Nội dung mục này đưa ra các khái niệm về kiểm thử dựa trên mô hình. Các quy trình về thực hiện kiểm thử dựa trên mô hình bắt đầu từ giai đoạn đặc tả.

Có nhiều khái niệm khác nhau về kiểm thử dựa trên mô hình. Tựu trung lại, chúng ta có thể hiểu kiểm thử dựa trên mô hình là một phương pháp kiểm thử nơi mà các ca kiểm thử được sinh ra từ mô hình đặc tả hành vi của hệ thống đang được kiểm thử. Mô hình này được biểu diễn bằng máy hữu hạn trạng thái, ôtômat, đặc tả đại số, biểu đồ trạng thái bằng UML, v.v.[1]

Chương 3: Phương pháp đặc tả tương tác giao diện cho các ứng dụng Web

Như chương 2 đã đề cập, phương pháp kiểm thử tự động tương tác giao diện cho các ứng dụng Web là phương pháp kiểm thử được sử dụng rộng rãi và phổ biến. Để có thể kiểm thử tự động được một ứng dụng Web bằng phương pháp này, trước hết chúng ta cần phải xây dựng hoặc đặc tả tương tác giao diện. Việc đặc tả cần áp dụng kiểm thử dựa trên mô hình. Lý thuyết về kiểm thử dựa trên mô hình cũng đã được chúng tôi đề cập trong chương 2.

3.1 Đặc tả tương tác giao diện của từng trang Web bằng ô-tô-mát hữu hạn trạng thái

Phương pháp đặc tả tương tác giao diện ứng dụng Web được chúng tôi sử dụng cho luận văn này là phương pháp đặc tả tương tác giao diện bằng ô-tô-mát hữu hạn trạng thái.

Dựa trên các yêu cầu và chức năng của hệ thống, có thể là các tài liệu thiết kế, kiểm thử viên đưa ra được mô hình dựa vào ô-tô-mát hữu hạn trạng thái. Với mục đích chính của luận văn mong muốn, kiểm tra tính đúng đắn của thiết kế so với chương trình.

Định nghĩa 3.1: Hành vi tương tác giao diện của một trang Web được đặc tả bằng ô-tô-mát trạng thái (Finite State Automaton - FSA) $M = < S, s_0, \sum, \delta, F >$

Chú ý 3.1: *Dạng <[điều kiện]sự kiện> có nghĩa là <sự kiện> chỉ xảy ra khi <điều kiện> được thỏa mãn. Ô-tô-mát hữu hạn trạng thái rỗng, ký hiệu là* $M = \prod$ *là ô-tô-mát hữu hạn trạng thái và có tập các trạng thái* $S = \emptyset$ *[3].*

Sau đây là một số khái niệm căn bản giúp chúng ta đặc tả một trang Web như là một ô-tô-mát hữu hạn trạng thái **Phần tử Web (Web Element),Trạng thái trang Web, Sự kiện.**

Trang Web mốc. Một trang Web $M_i = < S_i, s_{0i}, \sum_i, \delta_i, F_i >$ được gọi là trang Web mốc, nếu trang Web được dùng làm mốc để các trang Web khác ghép nối vào, hay nó là trang Web khởi đầu của ứng dụng Web. $M = < S, s_0, \sum, \delta, F >$ là trang Web sau khi ghép nối, với $s_0 = s_{0i}$.

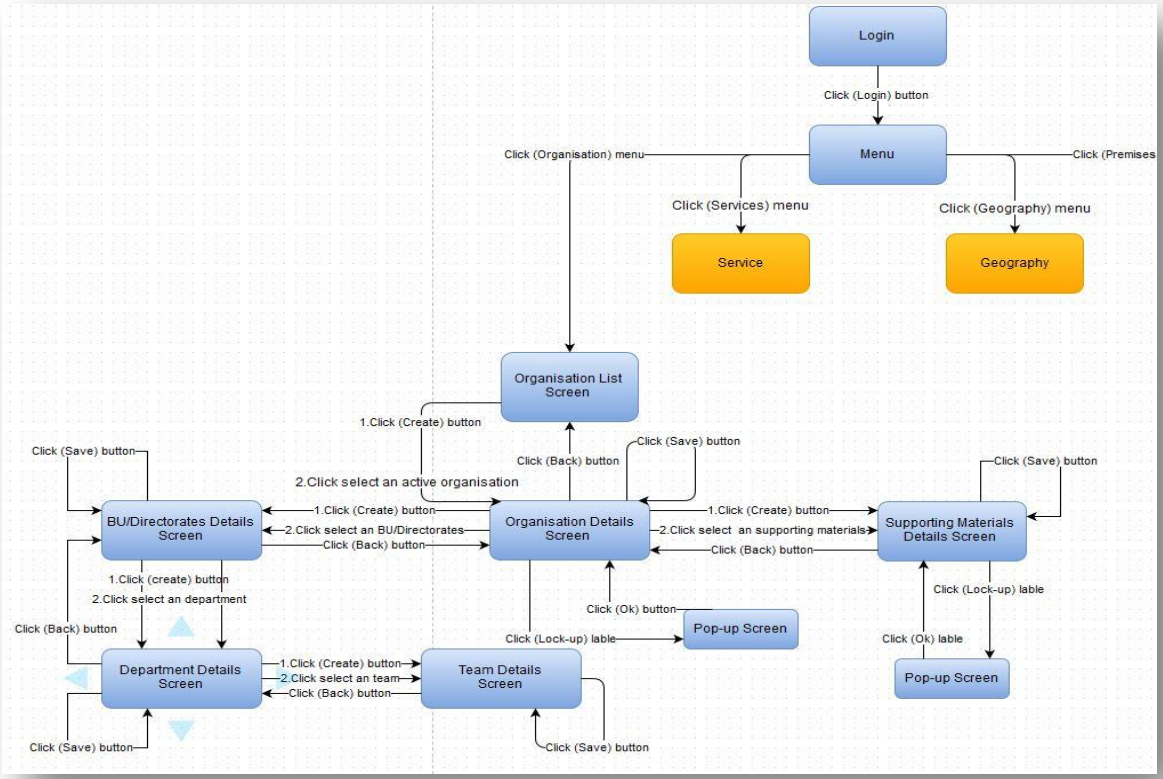
Mô hình của toàn bộ ứng dụng Web được xây dựng bằng cách ghép nối mô hình của tất cả các trang Web lại với nhau bằng phép toán ghép nối được định nghĩa [3] như sau:

Định nghĩa 3.2:

Chú ý 3.2: $M = \prod$ là ký hiệu ô-tô-mát hữu hạn trạng thái rỗng, tức nó có tập các trạng thái là rỗng.

3.2 Ví dụ minh họa cho đặc tả trang Web

Đầu vào của việc đặc tả tương tác giao diện cho ứng dụng Web là tài liệu thiết kế như hình 3.1. Dựa trên tài liệu thiết kế, ta có cái nhìn tổng quan về ứng dụng Web, từ đó xây dựng đặc tả cho từng trang Web. Theo như mô tả tại hình 3.1 ứng dụng Web bao gồm các trang:



Hình 3.1. Tài liệu thiết kế màn hình của dự án

Hình 3.2. Menu chính sau khi đăng nhập vào hệ thống

Hình 3.3. Trạng thái bắt đầu của Danh sách Tổ Chức (Organisation List)

Hình 3.4. Trạng thái Chi tiết về Tổ Chức (Organisation Details)

3.3.1 Xây dựng ô-tô-mát hữu hạn trạng thái M1

Để xây dựng được mô hình cho trang Web này, chúng ta cần xác định chính xác các trạng thái và sự kiện của nó. Các trạng thái của trang Danh Sách Tổ Chức (Organisation List) (Bảng 3.1.) được xác định dựa trên các trạng thái của các phần tử Web sau: tiêu đề trang Web (*Lable*), link của tổ chức (linkText)

Hình 3.5. Các trạng thái Web của trang Danh sách Tổ Chức

(Organisation List)

Stt	Tên trạng thái	Ý nghĩa
1	MenuList	Trạng thái ban đầu trước khi chọn Organisation List
2	OrgList	Khi người dùng chọn một tổ chức trong danh sách tổ chức để hiển thị thông tin
3	OrgDetails	Trạng thái trang Web khi trả về thông tin chi tiết của một tổ chức, và cũng là trạng thái kết thúc

Các sự kiện người dùng tương tác lên trang Web Danh sách Tổ Chức bảng 3.2

Hình 3.6. Các sự kiện của trang Danh sách Tổ Chức

Stt	Tên sự kiện	Ý nghĩa
1	select_MenuOrg	Người dùng chọn menu Organisation để hiển thị
2	click_Orgs	Người dùng lựa chọn 1 trong những Organisation trong danh sách để hiển thị chi tiết

Trang Danh sách Tổ chức (Organisation List) được đặc tả bằng ô-tô-mát hữu hạn trạng thái $M_1 = < S_1, s_{01}, \sum_1, \delta_1, F_1 >$ *(Hình 3.5)*, dựa trên tập trạng thái và tập sự kiện đã được nêu trên.

Hình 3.7. Ô-tô-mát hữu hạn trạng thái M1

Sau khi người dùng chọn một tổ chức để hiển thị, giao diện của trang Web Danh sách Tổ chức (Organisation List) sẽ chuyển sang trạng thái thông tin chi tiết của tổ chức (*OrgDetails*).

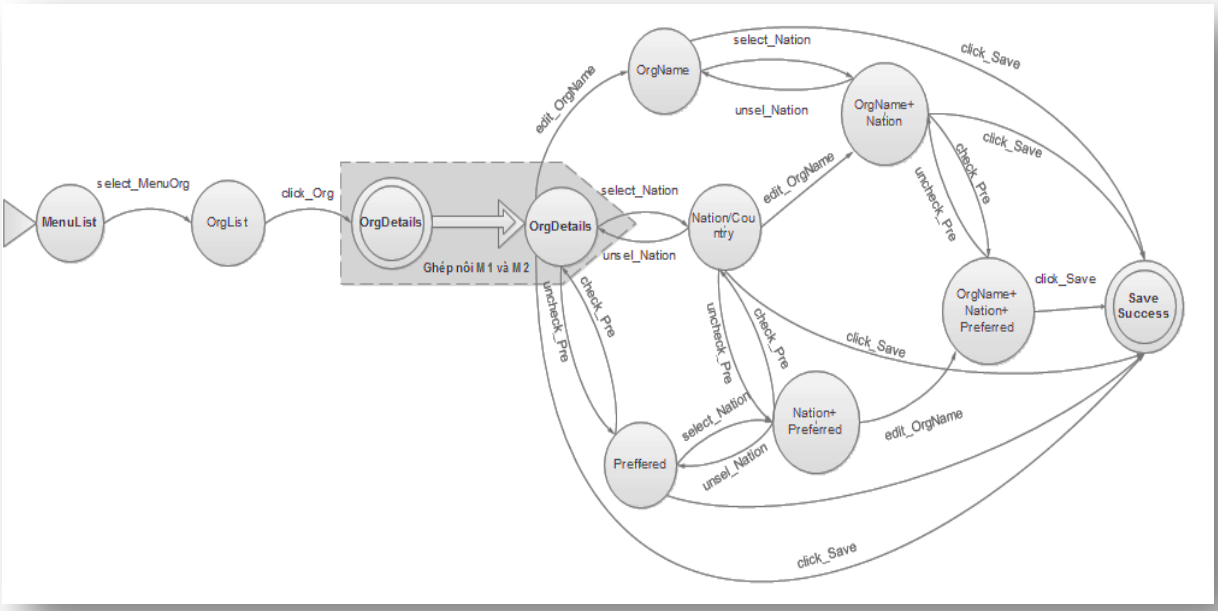
3.3.2 Ghép nối ô-tô-mát hữu hạn trạng thái M1 và M2

Trong mục này, ta sẽ thực hiện ghép nối ô tô mát hữu hạn trạng thái M1 và M2 dựa vào Website FPT Service.

Hình 3.8. Thông tin Chức Năng Sửa Organisation - Tab Information

Hình 3.9. Ô-tô-mát hữu hạn trạng M2

Để minh họa cụ thể cho phép toán ghép nối, chúng ta tiến hành ghép nối hai ô-tô-mát hữu hạn trạng thái của trang tìm kiếm cán bộ (search) và trang cập nhật thông tin cán bộ (update), như ở ví dụ 2.2 lại thành ô-tô-mát hữu hạn trạng thái $M = M_1 \| M_2 = < S, s_0, \sum, \delta, F >$, trong đó:



Hình 3.10. Mô hình M sau khi thực hiện thuật toán ghép nối giữa M1 và M2

Nhận xét: Phép ghép nối như định nghĩa trên có tính kết hợp nhưng không có tính giao hoán [3].

3.3 Biểu diễn mô hình đặc tả dưới dạng các tệp tin MS Excel

Công cụ để thực hiện sinh ca kiểm thử tự động cần phải có đầu vào là mô hình, nhưng công cụ không thể (hoặc chưa thể) đọc được trực tiếp từ mô hình ô-tô-mát hữu hạn. Do đó, trong đề tài luận văn này, người viết đã chuyển đổi mô hình ô-tô-mát dưới dạng các tệp tin Excel. Định dạng của tệp tin Excel được chia thành 4 bảng: *Element_html, State, Event, Transition.*

Tên bảng	**Mô tả**
Element_html	Mô tả các phần tử Web của trang Web
State	Mô tả các trạng thái
Event	Mô tả các sự kiện
Transition	Mô tả các hàm chuyển trạng thái giữa các trạng thái trong bảng *State*.
Số phần tử	Trên mỗi bảng đều có một ô đầu tiên số các phần tử của bảng

Sau đây, chúng tôi sẽ mô tả chi tiết từng bảng trong tệp tin Excel của trang cập nhật tổ chức *Organisation Details*

Bảng Element_html (bảng các phần tử Web): Bảng này được thiết kế để đặc tả các phần tử Web của trang Web cần kiểm thử.

Hình 3.11. Bảng các phần tử Web của trang Organisation Details, tab Information

Bảng 3.11 là bảng mô tả các phần tử Web của trang Organisation Details - tab Information. Cấu trúc của bảng này gồm có 4 cột chính: cột *id*, *html*, *type* và *value*. Mỗi cột của bảng mang ý nghĩa sau:

Tên bảng	**Mô tả**
id	Định danh
html	Lưu các thuộc tính của phần tử Web
type	Cột lưu kiểu của phần tử Web, công cụ có thể tiến hành kiểm thử thành công với đa số các kiểu phần tử Web
value	Cột lưu nội dung đầu ra mong muốn của phần tử Web
Số phần tử	Trên mỗi bảng đều có một ô đầu tiên số các phần tử của bảng

Bảng State (bảng trạng thái): là bảng mô tả các trạng thái của trang Web

Hình 3.12. Bảng các trạng thái của trang Organisation Details - tab Information

Bảng Event (Sự kiện): Bảng liệt kê tất cả các sự kiện diễn ra trên trang Web. Mỗi sự kiện là một tương tác người dùng lên một phần tử Web.

Hình 3.13. Bảng các sự kiện của trang Organisation Details - tab Information

Tên bảng	**Mô tả**
name	Tên sự kiện/hành vi tác động lên phần tử Web
html	Lưu các thuộc tính của phần tử Web cần tương tác. (cách thức đặc tả giống như cột *html* của bảng *Element_html*)
action	Hành vi của sự kiện, bao gồm: *click* (nhấn), *edittext(xóa chuỗi kí tự cũ và chèn kí tự mới), addtext* (thêm một chuỗi ký tự), *deltext* (xóa một chuỗi ký tự), *select* (chọn).

Bảng *Transition* (Sự chuyển trạng thái): Bảng này chứa toàn bộ các *transition* của trang Web.

Hình 3.14. Bảng các transition của trang Organisation Details - Tab Information

Mối liên kết giữa các bảng: Tệp tin Excel đặc tả trang Web tìm kiếm cán bộ gồm 4 bảng trên, 4 bảng này được nối lại như bảng 3.15.

Hình 3.15. Tệp tin Excel đặc tả trang Web Organisation Details - Tab Information

Một tiện ích hỗ trợ đã được giới thiệu tại [3]. Ngoài ra còn một số yêu cầu về thiết kế trang Web nhằm phục vụ cho việc đặc tả ứng dụng Web bằng phương pháp đã nêu trên như sau.

Chương 4: Sinh và thực thi các ca kiểm thử tự động

Chương 3 đã trình bày phương pháp đặc tả hình thức tương tác giao diện ứng dụng Web như là một ô-tô-mát hữu hạn trạng thái và được biểu diễn dưới dạng các tệp tin Excel. Chương này, chúng tôi sẽ trình bày hai thuật toán sinh tự động các ca kiểm thử từ phương pháp đặc tả mô hình trên.

4.1 Sinh các ca kiểm thử từ mô hình đặc tả hình thức

4.1.1 Đường dẫn kiểm thử

Trong nghiên cứu này, các đường dẫn kiểm thử được sinh ra từ thuật toán có cấu trúc như sau: <*trạng thái bắt đầu*> * <*sự kiện*$_i$> = <*trạng thái*$_i$> * ... = <*trạng thái kết thúc*> [3]. Đường dẫn bắt đầu từ trạng thái bắt đầu của hệ thống, nếu sự kiện <*sự kiện*$_i$> xảy ra thì hệ thống sẽ chuyển sang trạng thái tiếp theo <*trạng thái*$_i$>. Tiếp tục cho đến hết, trạng thái cuối cùng của đường dẫn chính là trạng thái kết thúc của hệ thống.

4.1.2 Thuật toán sinh tự động các đường dẫn kiểm thử

Khi chúng ta đã xây dựng được mô hình đặc tả chính xác hành vi của hệ thống, một trong những công việc khó khăn còn lại là làm thế nào để sinh được các ca kiểm thử từ mô hình này [1]. Để làm việc này, với mỗi mô hình được đặc tả bằng FSA, chúng ta có thể coi nó như là một đồ thị chuyển trạng thái và áp dụng một trong các phương pháp: duyệt ngẫu nhiên, duyệt theo chiều sâu hoặc duyệt theo chiều rộng trên đồ thị đó, thông qua các trạng thái và các chuyển trạng thái giữa chúng. Một đường đi từ trạng thái khởi tạo đến một trạng thái kết thúc tương ứng với một ca kiểm thử chúng ta muốn tạo ra [1].

Thuật toán duyệt theo chiều sâu

Algorithm 1 *Depth_First_Search*: int i, path PATH

```
1: // PATH là biến lưu các test path
2: // arr lưu test path tạm thời
3: Bool backTrack = true;
4: for tất cả các cạnh do
5:    if  cạnh chưa được duyệt then
6:       backTrack = false;
7:       Thông báo cạnh đã duyệt qua;
8:       Thêm đỉnh vào arr;
9:       Depth_First_Search: j, PATH;
10:      Bỏ đỉnh khỏi arr
11:   end if
12: end for
13: if backTrack = true then
14:    Thêm arr vào PATH;
15: end if
```

Hình 4.1. Thuật toán duyệt đồ thị theo chiều sâu.

Thuật toán add thêm Path để bổ sung trạng thái cuối cho đường dẫn kiểm thử.

Algorithm 2 *ADD_Path*: path *PATH*

```
1: loop
2:   for duyệt các test path của PATH do
3:     if trạng thái cuối cùng của test path i không là trạng thái kết thúc then
4:       while trạng thái cuối cùng của test path i không là trạng thái kết thúc do
5:         for duyệt tất cả các cạnh của đồ thị do
6:           if cạnh đồ thị có đỉnh bắt đầu là đỉnh cuối của PATH then
7:             Thêm đỉnh cuối của cạnh vào test path i;
8:           end if
9:         end for
10:      end while
11:    end if
12:  end for
13: end loop
```

Hình 4.2. Thêm các đường cạnh vào đường dẫn kiểm thử.

Áp dụng hai thuật toán trên chúng ta có thể sinh ra tất cả các đường dẫn kiểm thử từ mô hình tương tác của người dùng trên giao diện Web. Chất lượng của các đường dẫn kiểm thử này phụ thuộc vào mô hình được đặc tả trước đó.

4.1.3 Ví dụ áp dụng thuật toán

Áp dụng thuật toán 1 và thuật toán 2 đã nêu ở phần trước cho ô-tô-mát hữu hạn trạng thái của trang Organisation Details - tab Information Hình 4.1.

Bảng 4.1. Các transition của trang Organisation Details - tab Information

Bảng 4.2. Các testpath được sinh ra từ mô hình trang Organisation Details - tab Information

Kết quả sau khi duyệt đồ thị đã thỏa mãn các điều kiện sau:

- Đảm bảo tất cả các cạnh đều đã được duyệt qua
- Mọi test path đều bắt đầu bằng trạng thái khởi đầu của đồ thị.
- Trạng thái cuối cùng của mỗi test path là trạng thái nằm trong tập trạng thái kết thúc của đồ thị.

Như vậy, chúng ta đã sinh ra tất cả các đường dẫn kiểm thử từ mô hình ô-tô-mát hữu hạn trạng thái của trang tìm kiếm thông tin cán bộ.

4.2 Thực hiện các ca kiểm thử

Chúng ta sẽ thực thi các ca kiểm thử được sinh ra từ thuật toán trên nhằm phát hiện các lỗi lập trình so với đặc tả (mô hình hệ thống).

- Bước 1: Tách đường dẫn kiểm thử thành các đường dẫn kiểm thử nhỏ (*transition*).
- Bước 2: Thực hiện từng đường dẫn nhỏ. Phương pháp sẽ xác định *trạng thái*$_i$ của trang Web. Nếu xác định trạng thái thành công, tiếp tục dùng hai công cụ bổ trợ trên để xác định vị trí phần tử Web và thực hiện sự kiện. Cuối cùng, xác định trạng thái của trang Web sau khi đã thực hiện sự kiện, bằng cách lấy các giá trị của các phần tử Web tương ứng với trạng thái này (trạng thái Web thực tế),

so sánh với giá trị trạng thái của trang Web trong bản đặc tả (trạng thái Web mong muốn) và chuyển sang bước 3. Ngược lại thì dừng thực hiện và chuyển sang bước 4.

- Bước 3: Nếu kết quả so sánh trên trùng nhau thì tiếp tục lặp lại hai bước trên cho đường dẫn con tiếp theo. Ngược lại, thì kết thúc việc thực hiện đường dẫn lớn và chuyển sang bước 4.
- Bước 4: Kết luận về kết quả thực hiện đường dẫn kiểm thử.

4.3 Đánh giá phương pháp

Việc xây dựng một mô hình bằng ô-tô-mát hữu hạn trạng thái cũng vô cùng phức tạp. Số lượng các trạng thái cho các ứng dụng này có thể lên đến hàng nghìn hoặc nhiều hơn nữa, khó khăn khi biểu diễn ô-tô-mát hữu hạn trạng thái bằng đồ thị chuyển trạng thái.

Ngoài ra, việc đặc tả ứng dụng Web bằng ô-tô-mát hữu hạn trạng thái vẫn còn tồn tại một số vấn đề như: trạng thái có các hành vi khó xác định, thiếu trạng thái, thừa trạng thái. Khó khăn lớn nhất để có thể áp dụng phương pháp một cách hiệu quả là phải có một chuyên gia đặc tả hình thức có khả năng phân tích, thiết kế hệ thống và phải hiểu hệ thống thật chi tiết.

Đối với phương pháp sinh các đường dẫn kiểm thử bằng cách duyệt đồ thị thì đây là một phương pháp có khả năng bao phủ tốt các trường hợp cần kiểm thử và tương đối dễ dàng cài đặt. Nhưng có một vấn đề trong việc tìm kiếm các ca kiểm thử ở phương pháp này. Đó là, số lượng các đường dẫn kiểm thử được sinh ra là rất lớn, trong khi đó chỉ thực sự cần những đường dẫn kiểm thử có thể tìm được lỗi của ứng dụng Web.

Như vậy, để vận dụng và thực hiện một cách hiệu quả phương pháp kiểm thử dựa trên mô hình này thì chúng ta cần phải khắc phục một số khó khăn trên.

Chương 5: Công cụ và thực nghiệm

Chương 3, chương 4 đã trình bày phương pháp đặc tả hình thức giao diện ứng dụng Web bằng ô-tô-mát hữu hạn trạng thái, phương pháp sinh tự động các ca kiểm thử bằng cách duyệt đồ thị trạng thái. Nội dung của chương 4 sẽ trình bày về công cụ kiểm thử tự động tương tác giao diện các ứng dụng Web được xây dựng từ các phương pháp trong chương 2, 3 và trình bày kết quả thực nghiệm.

5.1 Giới thiệu các công cụ bổ trợ

5.1.1. Giới thiệu Selenium và một số API WebDriver được sử dụng

Selenium được chia làm bốn phần:

- Selenium IDE
- Selenium RC (Selenium 1 – Selenium Remote Control)
- Selenium Gird
- Selenium WebDriver (Selenium 2)

Hình 5.1. Cấu trúc của Selenium [1]

Hiện nay, Selenium được sử dụng khá nhiều trong việc kiểm thử tự động các ứng dụng Web.Theo tài liệu [7] Selenium được giới thiệu với các đặc tính nổi trội và linh hoạt như bên dưới:

1) Mã nguồn mở.
2) Cộng đồng hỗ trợ.
3) Selenium hỗ trợ nhiều ngôn ngữ lập trình.
4) Selenium hỗ trợ chạy trên nhiều OS khác nhau với mức độ chỉnh sửa script hầu như là không có. Thực sự thì điều này phụ thuộc phần lớn vào khả năng viết script của chúng ta.
5) Chạy test case ở backround.
6) Không hỗ trợ Win app.

Định nghĩa về từng phần của Selenium như hình 5.1 : Selenium IDE, Selenium RC, Selenium Grid, Selenium Webdriver (Selenium2)

Một số Selenium-WebDriver API được sử dụng để kết nối với ứng dụng Web và thực hiện các ca kiểm thử của ứng dụng Web đó, bao gồm:

API kết nối đến một trình duyệt Web: API Selenium này giúp kết nối đến một đối tượng trình duyệt Web. Câu lệnh kết nối đến một trình duyệt Web FireFox.

```
WebDriver firefoxDriver = new FirefoxDriver();
firefoxDriver.get("http://www.google.com");
firefoxDriver.navigate.to("http://www.gooogle.com");
```

Driver.close(), Driver.quit()

API xác định vị trí các phần tử Web: Theo [9,10] Selenium hỗ trợ các kiểu bộ định vị sau: id, name, xpath, dom, identifier, link, css

Có nhiều các phương thức khác nhau để xác định vị trí phần tử Web. Người dùng có thể lựa chọn một trong các phương thức bên dưới phù hợp với từng điều kiện và ứng dụng.

Xác định vị trí trả về một giá trị	Xác định vị trí trả về một danh sách
(1) find_element_by_id (2) find_element_by_name (3) find_element_by_xpath (4) find_element_by_link_text (5) find_element_by_partial_link_text (6) find_element_by_tag_name (7) find_element_by_class_name (8) find_element_by_css_selector	(1) find_elements_by_name (2) find_elements_by_xpath (3) find_elements_by_link_text (4) find_elements_by_partial_link_text (5) find_elements_by_tag_name (6) find_elements_by_class_name (7) find_elements_by_css_selector

Thao tác với các phần tử Web: Sau khi Selenium xác định được vị trí các phần tử Web. Chúng ta sẽ thực hiện các thao tác lên phần tử Web đó thông qua một số hàm được Selenium-WebDriver cung cấp nhằm thực hiện các thao tác chuột và bàn phím đối với phần tử Web. Theo [11] có tổng hợp tất cả các thao tác được sử dụng trong Selenium Webdriver như Bảng 5.1.

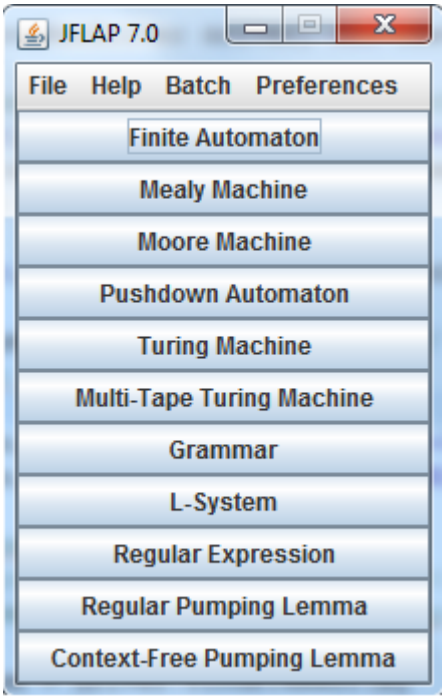
Bảng 5.1. Các thao tác lên phần tử Web

5.1.2. Công cụ JFLAP

Công cụ JFLAP được sử dụng như một tiện ích mở rộng phục vụ cho công việc đặc tả các hành vi của hệ thống dựa trên mô hình. JFLAP là một phần mềm mã nguồn mở, trong đó người dùng có thể biểu diễn các ô-tô-mát hoặc các máy hữu hạn trạng thái. Trong phần này, chúng tôi sẽ mô tả về lợi ích khi sử dụng công cụ JFLAP và biểu diễn cách sử dụng công cụ.[8]

5.1.2.1. Màn hình biểu diễn

Công cụ JFLAP thì hỗ trợ rất nhiều các chức năng. Tuy nhiên, nghiên cứu chỉ sử dụng chức năng Finite Automaton để thực hiện.



Sau khi chọn chức năng Finite Automata, màn hình chính xuất hiện với các thanh công cụ hỗ trợ thiết kế mô hình.

Bảng 5.2. Các công cụ trên JFLAP

5.1.2.2. Tạo các trạng thái và tạo các đường chuyển trạng thái

a. Tạo các trạng thái (State)

Đưa các hình vẽ để mô tả các thao tác thực hiện để tạo một trạng thái, và cách thay đổi trạng thái của các State. Xác định state là điểm đầu hay điểm kết thúc.

b. Tạo các đường chuyển trạng thái (Transition)

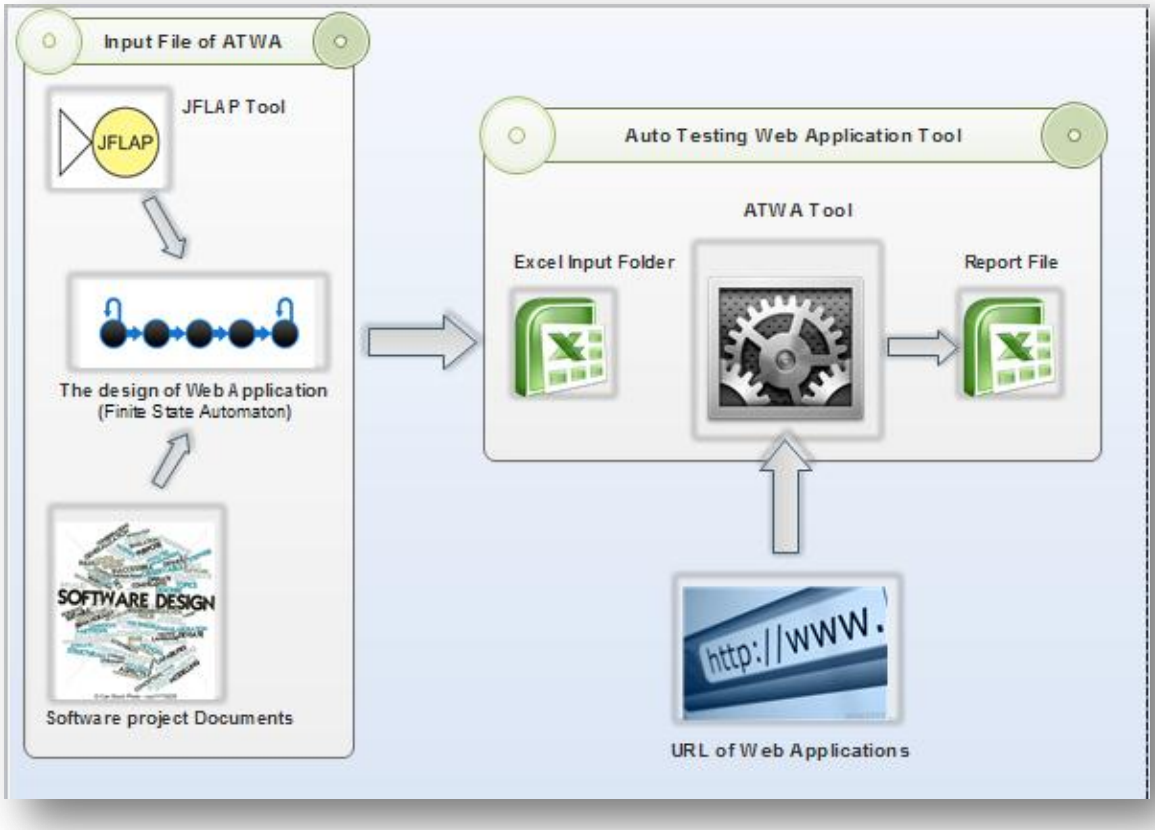
Đưa ra các hình ảnh và hướng dẫn về cách tạo các transition trong công cụ.

5.2 Giới thiệu công cụ kiểm thử tự động tương tác giao diện cho các ứng dụng Web

Chi tiết về kiến trúc của công cụ, đầu vào, và đầu ra của công cụ sẽ được mô tả rõ trong các mục bên dưới.

5.2.1 Kiến trúc của công cụ

Kiến trúc của công cụ kiểm thử tự động tương tác giao diện các ứng dụng Web (Auto Testing Web Application) được mô tả như trong hình 5.2.



Hình 5.2. Kiến trúc của công cụ Auto Testing Web Application

Công cụ được chia là hai phần chính, phần thứ nhất là đầu vào *Input File of ATWA* và phần xử lý của công cụ *Auto Testing Web Application Tool*. Công cụ sử dụng Selenium framework được tính hợp trong ATWA

5.2.2 Đầu vào của công cụ

5.2.2.1 Dựng mô hình và xuất tệp tin Excel bằng công cụ JFLAP

a. Dựng mô hình

Đối với mỗi một trang Web đều có những điểm đặc biệt khác nhau. Điểm chung duy nhất cần để dựng mô hình là cần phải nắm rõ luồng hoạt động của các trang Web, chúng ta có thể tham khảo tài liệu thiết kế các màn hình và luồng hoạt động ví dụ như Hình 5.3.

Hình 5.3. Ví dụ về luồng của các màn hình trong trang Web

b. Xuất tệp tin Excel

Theo như công cụ phát triển tại [3], công cụ JFLAP đã thực hiện phát triển thành công việc xuất tệp tin excel đầu vào của công cụ.

- **Quy tắc đặt tên cho trạng thái và đường chuyển trạng thái**

Cách định nghĩa trạng thái
Tên_trạng_thái|element_html_id|kiểu_phần_tử
Ví dụ: UserName|txt_UserName|textbox

Bảng 5.3. Bảng Element_html

Bảng 5.4. Bảng Sate (bảng trạng thái)

Cách định nghĩa sự kiện
Tên_sự_kiện|element_html_id|hành_vi
Ví dụ: add_UserName|txt_UserName|addtext

Bảng 5.5. Bảng Event (bảng sự kiện)

- **Xuất tệp tin excel**

Người thiết kế sau khi đã tạo máy trạng thái thành công, sử dụng công cụ JFLAP ta có thể xuất nội dung ra tệp tin *excel* cho thiết kế đầu vào của công cụ kiểm thử tự động Web. Chúng ta lấy một ví dụ để minh họa cho việc xuất tệp tin excel như Hình 5.5

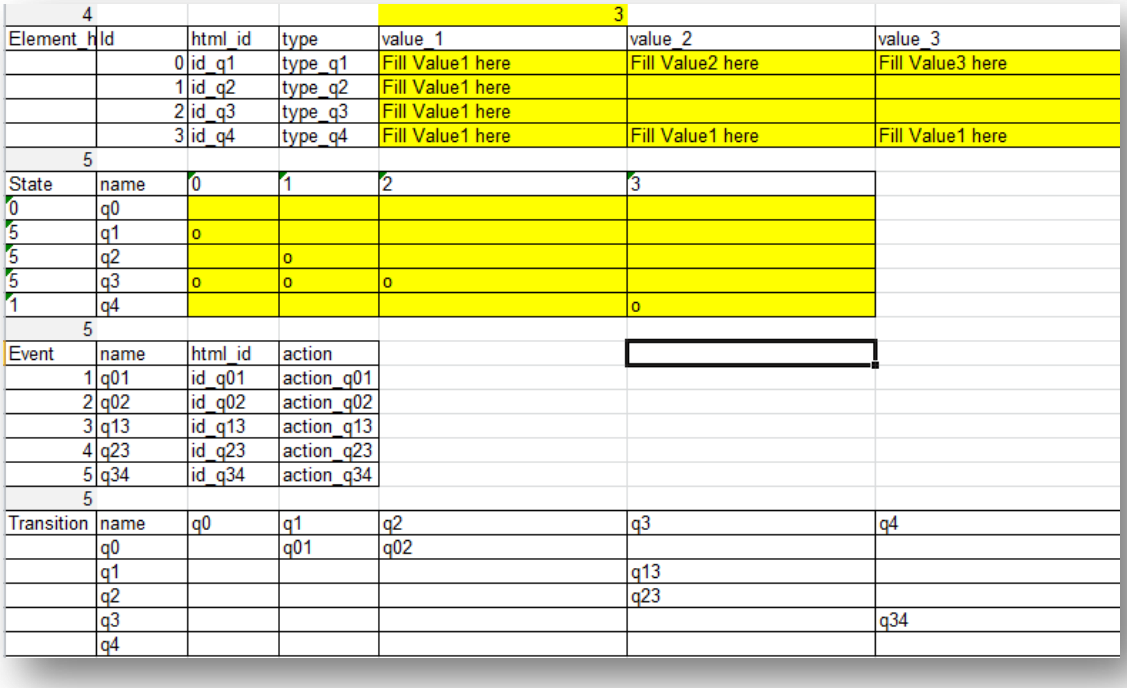
Hình 5.4. Ví dụ về một FA với cách định nghĩa như mục a.

Hình 5.5. Xuất ra tệp tin excel.

Sau khi thực hiện xuất ra tệp tin excel từ mô hình Hình 5.6., tệp tin excel được hiển thị như Hình 5.7.

Hình 5.6. Tệp tin excel sau khi được xuất từ công cụ JFLAP

Như vậy, tệp tin đầu vào của công cụ đã gần như hoàn chỉnh chỉ cần bổ sung thêm các giá trị cần kiểm thử Hình 5.8. (phần bôi vàng)



4				3		
Element_h	Id	html_id	type	value_1	value_2	value_3
	0	id_q1	type_q1	Fill Value1 here	Fill Value2 here	Fill Value3 here
	1	id_q2	type_q2	Fill Value1 here		
	2	id_q3	type_q3	Fill Value1 here		
	3	id_q4	type_q4	Fill Value1 here	Fill Value1 here	Fill Value1 here
5						
State	name	0	1	2	3	
0	q0					
5	q1	o				
5	q2		o			
5	q3	o	o	o		
1	q4				o	
5						
Event	name	html_id	action			
1	q01	id_q01	action_q01			
2	q02	id_q02	action_q02			
3	q13	id_q13	action_q13			
4	q23	id_q23	action_q23			
5	q34	id_q34	action_q34			
5						
Transition	name	q0	q1	q2	q3	q4
	q0		q01	q02		
	q1				q13	
	q2				q23	
	q3					q34
	q4					

Hình 5.7. Tệp tin excel đầu vào sau khi được điền giá trị kiểm thử

5.2.2.2 Cách đặt tên và dựng tệp tin Excel

Ví dụ, Hình 5.3, cách đặt tên của file excel được đặt theo số thứ tự và đưa vào các thư mục mà cần kiểm thử riêng lẻ.

Hình 5.8. Lưu trữ các tệp tin đầu vào

5.2.2.3 Giao diện và cách sử dụng công cụ ATWA

Dựa vào Hình 5.9 và 5.10 để hướng dẫn về cách sử dụng công cụ ATWA

Hình 5.9. Giao diện nhập dữ liệu đầu vào của công cụ

Hình 5.10. Chọn bộ kiểm thử để thực hiện

Công cụ sử dụng Selenium Webdriver để kết nối nên trong góc bên phải của thanh công cụ sẽ hiển thị chữ Webdriver.

Hình 5.11.Selenium Webdriver thực hiện kiểm thử tự động trên Web

Và cuối cùng, sau khi thực hiện xong, kết quả được hiển thị như Hình 5.11 và 5.12

Hình 5.12.Kết quả hiển thị sau khi chạy xong bộ kiểm thử

5.2.3 Đầu ra của công cụ

Sau khi thực hiện chọn và kiểm thử tự động (như hướng dẫn tại 5.2.3) công cụ sẽ xuất một tệp tin Excel có mô tả kết quả kiểm thử PASS hoặc FAIL

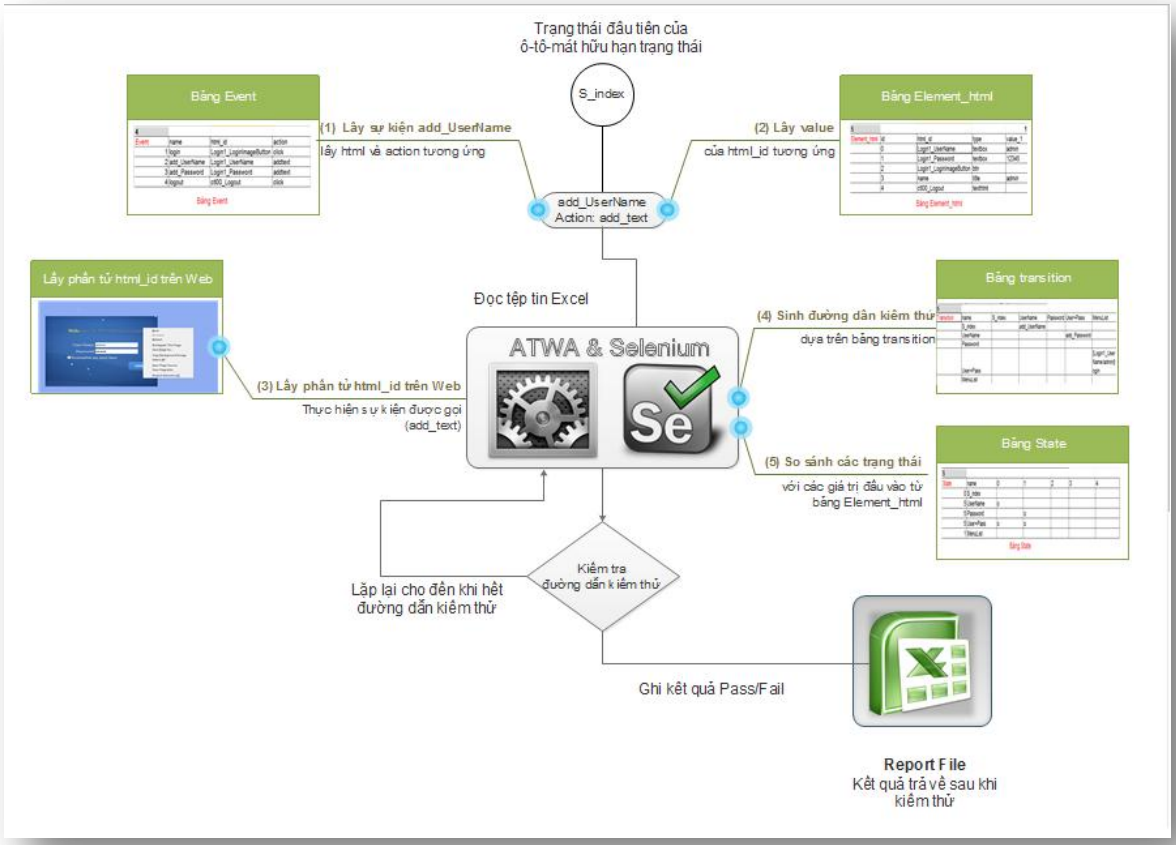
Bảng 5.6. Các trường hợp thất bại *FAIL*

Ví dụ về đường dẫn kiểm thử sau khi ghép nối ba ô-tô-mát hữu hạn trạng thái.

Đường dẫn kiểm thử:

S_index**add_UserName=**UserName***add_Password=**User+Pass***login=**MenuList***click_Org MenuList=**OrgList***select_Org=**OrgDetails***edit_OrgName=**OrgName***click_Save=SaveSuccess*

Công cụ thực hiện từng *transition*: S*_index* * *add_Username* = *Username* như Hình 5.14. Ví dụ về hoạt động của cộng cụ ATWA để xuất ra tệp tin báo cáo.



Hình 5.13.Ví dụ về hoạt động của công cụ ATWA

Kết quả này sẽ được xuất ra tệp tin Report File như Hình 5.14

Hình 5.14.Kết quả kiểm thử.

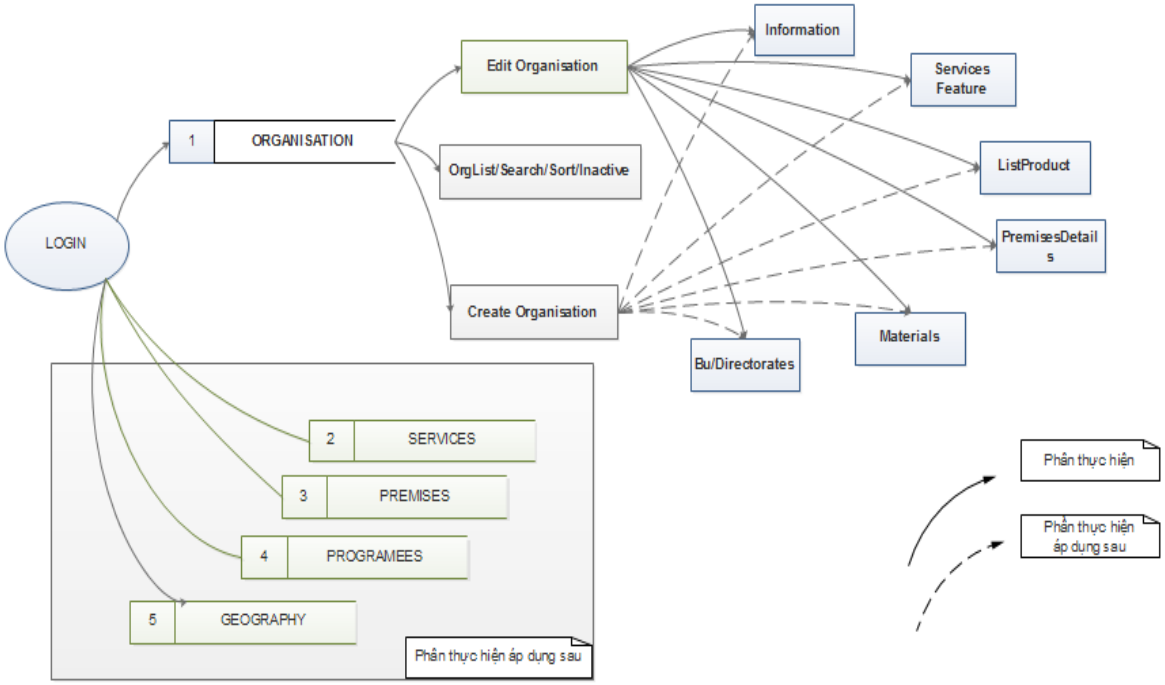
5.2.4 Thực nghiệm

5.2.4.1. Giao diện của ứng dụng Web

Trong mục này, tôi sẽ giới thiệu cách áp dụng công cụ kiểm thử tự động vào ứng dụng FPT-SD Web. Đây là một trang Web quản lý các tổ chức do FPT phát triển. FPT-SD viết tắt của từ FPT - Services Directory - kho lưu trữ dịch vụ của công ty FPT. Trang Web có các chức năng chính:

Bảng 5.7. Bảng chức năng chính của trang Web FPT Services

Ứng dụng Web này có cấu trúc các trang Web như Hình 5.15:



Hình 5.15. Ứng dụng Web quản lý thông tin cán bộ.

Luận văn chỉ xin đề cập đến phần sẽ thực hiện như đã nêu trong hình vẽ 5.15

Login (*Đăng nhập*): Trang thực hiện việc đăng nhập vào ứng dụng Web Hình 5.16 .

Hình 5.16. Giao diện trang đăng nhập

Hình 5.17.Danh sách các chức năng (Menu List)

Organisation (Quản lý tổ chức): Trang Web này có chức năng quản lý các tổ chức của công ty, trong đó có các chức năng chính như Bảng 5.9 mô tả

Bảng 5.8. Chức năng chính của Organisation

Hình 5.18.Giao diện các chức năng của Organisation.

Hình 5.19.Chức năng tìm kiếm theo bảng chữ cái

Hình 5.20. Chức năng sắp xếp Organisation

Hình 5.21.Giao diện các chức năng của Organisation.

Hình 5.22.Giao diện phần Service Features

Hình 5.23.Giao diện phần List Product

Hình 5.24.Giao diện phần Premise Detail

Hình 5.25.Giao diện phần Materials

Hình 5.26.Giao diện phần Bu Derectorate

5.2.4.2. Đặc tả mô hình Ô-tô-mát hữu hạn trạng thái

Từ giao diện được giao diện được mô tả tại mục 5.2.4.1 và mô hình ô-tô-mát hữu hạn trạng thái được chia làm ba mô hình lần lượt như các hình bên dưới.

Login

Hình 5.27.Mô hình ô-tô-mát hữu hạn trạng thái trang Login

Organisation Deatails

Hình 5.28. Mô hình ô-tô-mát hữu hạn trạng thái chức năng

Organisation Details

Edit Organisation

Hình 5.29.Mô hình ô-tô-mát hữu hạn trạng thái chức năng

Edit Organisation tại Tab Infomation

5.2.4.3. Tệp tin đầu vào

Hình 5.30 là thư mục chứa các bản đặc tả tương tác giao diện của chức năng Organisation. Thư mục này gồm 3 tệp tin Excel đặc tả tương tác giao diện của 6 trang Web tương ứng như các hình 5.27, 5.28, 5.29

(1) Trang đăng nhập - trang Web được chọn làm mốc *0-login.xls*

(2) Trang chi tiết tổ chức Organisation Details *2-OrgDetails.xls*

(3) Trang Sửa Organisation với tab Information *3-OrgInform.xls*

Hình 5.30.Thư mục các tệp tin đặc tả chức năng Organisation

5.2.4.4. Kết quả đầu ra

Nêu từng bước để xuất ra tệp tin excel đầu ra.

Hình 5.31.Giao diện của công cụ

Hình 5.32.Các đường dẫn kiểm thử được sinh tự động

Hình 5.33.Kết quả thực hiện đường dẫn kiểm thử hiển thị trong tệp tin đầu ra

5.2.5 Kết quả áp dụng và cải tiến công cụ

Thực hiện phát triển công cụ dựa vào [3] và [2].

Công cụ [2] của Lê Thị Phượng có cải tiến về code

Kết quả áp dụng

Áp dụng thành công trên Website của công ty FPT.Đặc biệt áp dụng vào giai đoạn Kiểm thử chấp nhận hoặc kiểm thử hệ thống. So sánh số lượng test case giữa kiểm thử normal và test case sinh ra, và số lỗi tìm được Các report báo cáo về giai đoạn kiểm thử chấp nhận, nhấn mạnh kiểm thử trong mô hình Agile hiện nay (Kiểm thử chấp nhận như là kiểm thử hồi quy)

Cải tiến công cụ : Cải tiến công cụ JFLAP tạo tệp tin đầu ra. Code tách riêng các hàm sự kiện, đối với từng sự kiện thì sẽ thự hiện theo một hàm riêng Thực hiện cấy lỗi cho ứng dụng thì công cụ có tìm ra lỗi.

5.2.6 Ý nghĩa của công cụ thực nghiệm

.

Chương 6: KẾT LUẬN

Như đã trình bày trong Chương 1, kiểm thử tự động dường như trở thành một xu hướng tất yếu trong việc phát triển phần mềm nhanh, mô hình Agile hay Scrum là một ví dụ. Chúng ta không thể phủ nhận việc áp dụng kiểm thử tự động làm cho hiệu quả công việc cao hơn, giảm thiểu rủi ro trong sai sót, và giảm chi phí.

Kiểm thử tự động tương tác giao diện người dùng được xem như một giải pháp rất hữu hiệu khác với những kiểm thử tự động thường sử dụng là kiểm thử hiệu năng, kiểm thử bảo mật, kiểm thử chịu tải.v.v. Kiểm thử tương tác giao diện người dùng sẽ cho hiệu quả khác biệt khi thực hiện một cách tự động việc kiểm tra tính đúng đắn của chương trình so với các tài liệu thiết kế và các tài liệu đặc tả ban đầu. Không những thế, kiểm thử tự động tương tác giao diện người dùng còn giúp kiểm thử nhanh và chuẩn xác. Luận văn này được áp dụng và phát triển dựa trên đề xuất của [2] và [3]. Đề xuất của [2] và [3] đã đưa ra được các tính năng sau: kiểm thử luồng giao diện, xác định được phần tử định danh, xác định các phần tử có kiểu *name, class* , giao diện popup, kiểm thử được các loại phần tử Web như Dropdownlist, Checkboxlist, RadioList, v.v.

Tuy nhiên, dựa vào công cụ [3] và cải tiến của [2] thì việc kiểm thử tự động còn gặp nhiều khó khăn trong khâu thiết kế đầu vào, tạo các tệp tin excel, đặc biệt đối với những người lần đầu sử dụng sẽ rất khó và không tránh khỏi sai sót. Luận văn trình bày trên ngoài việc áp dụng thành công cải tiến của [3] và [2] đã cải tiến việc tạo tệp tin đầu vào bằng việc phát triển công cụ JFLAP. Đối với [2] tệp tin đầu vào được tạo bằng cách thủ công. Đối với [3], thì công cụ JFLAP đã được đưa vào sử dụng, nhưng mới chỉ dừng tại việc xuất ra một tệp tin excel chỉ có duy nhất 1 bảng, việc tạo thủ công cho phần tiếp theo cũng gặp hạn chế, dễ xảy ra sai sót.

Phương pháp kiểm thử tự động dựa vào hành vi tương tác giao diện ứng dụng Web đã trình bày có các ưu điểm nổi bật như: chỉ phụ thuộc vào phần hiển thị HTML mà không phụ thuộc vào ngôn ngữ lập trình Web, kiểm thử được cho hầu hết các loại phần tử Web và dễ dàng tạo các tệp tin đầu vào giúp giảm chi phí và công sức thực hiện. Với những ưu điểm trên, phương pháp hứa hẹn sớm được áp dụng rộng rãi trong thực tế, trở thành một công cụ hữu hiệu cho việc kiểm thử các ứng dụng Web hiện nay.

Về thực nghiệm, chúng tôi đã áp dụng công cụ kiểm thử tự động tương tác giao diện các ứng dụng Web cho một số hệ thống tại FPT Software. Dựa vào kết quả thực nghiệm, công cụ đã cho chúng ta thấy được tính khả dụng cũng như khả năng tạo những đường dẫn kiểm thử bao phủ được hầu hết các trường hợp có thể xảy ra trong hệ thống.

Trong tương lai, chúng tôi sẽ áp dụng công cụ cho các hệ thống phức tạp hơn nhằm chứng minh tính hiệu quả của phương pháp. Đồng thời, chúng tôi sẽ khắc phục và cải tiến công cụ để có thể kiểm thử tự động trên các trình duyệt có version cao hơn. Cải thiện công cụ JFLAP để tích hợp với công cụ ATWA để có thể đọc luôn từ mô hình mà không cần phải thông qua tệp tin excel tiến tới một công cụ mang ý nghĩa tự động một cách hoàn toàn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Phạm Ngọc Hùng, Trương Anh Hoàng, Đặng Văn Hưng (2014), *Giáo trình kiểm thử phần mềm*, NXB Đại học Quốc gia Hà Nội.

[2] Lê Thị Phượng (2015), Nghiên cứu về kiểm thử dựa trên mô hình và ứng dụng, Luận văn thạc sĩ, Trường Đại Học Công Nghệ, Đại học Quốc Gia Hà Nội.

Tiếng Anh

[3] Khanh Trinh Le, Hieu Dinh Vo and Pham Ngoc Hung, (2015), "A Method for Automated User Interaction Testing of Web Applications", Journal on Information and Communications Technology (JoICT)

[4] Mark Fewster , Dorothy Graham (2006), "Software Test Automation", The First Combined International Conference on Formal Approaches to Software Testing and Runtime Verification, FATES'06/RV'06, pp. 115–132.

[5] Ann Millikin, (2014), *What Types of Software Testing Can and Should Be Automated*, http://www.seguetech.com/.

[6] Ana Cristina Ramada Paiva Pimenta, (2006), *Automated Specification-Based Testing of Graphical User Interface*, Thesis of Porto University, Portugal.

[7] Selenium Document Team, (2010), "Selenium Documetation version 1.0", *Note to the reader on Website http://www.seleniumhq.org/docs*

[8] JFLAP Tutorial, Guideline for user on Website http://jflap.org/tutorial/

[9] Baiju Muthukadan, (2016), "Selenium Python Bindings", 2, Guideline from http://selenium-python.readthedocs.org/, pp.15-20

[10] Selenium-WebDriver API Commands and Operations, Guideline for user on Website http://www.seleniumhq.org

[11] Vineet Kumar, (2015), *Action Class in Selenium*, Journal on Website http://www.seleniumwebdriver.in

Jinlin Chen, Baoyao Zhou, Jin Shi, Hongjiang Zhang, Qiu Fengwu, (2001), "Function-Based Object Model Towards Website", Hong Kong